



## Subject card

Subject name and code	Software Engineering, PG_00047826						
Field of study	Informatics						
Date of commencement of studies	October 2020		Academic year of realisation of subject		2021/2022		
Education level	first-cycle studies		Subject group		Obligatory subject group in the field of study Subject group related to scientific research in the field of study		
Mode of study	Part-time studies		Mode of delivery		at the university		
Year of study	2		Language of instruction		Polish		
Semester of study	4		ECTS credits		4.0		
Learning profile	general academic profile		Assessment form		exam		
Conducting unit	Department of Software Engineering -> Faculty of Electronics, Telecommunications and Informatics						
Name and surname of lecturer (lecturers)	Subject supervisor		dr inż. Aleksander Jarzębowicz				
	Teachers		dr inż. Maciej Kucharski				
Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	15.0	0.0	15.0	0.0	0.0	30
	E-learning hours included: 0.0						
	Adresy na platformie eNauczanie:						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan		Participation in consultation hours		Self-study	SUM
	Number of study hours	30		4.0		66.0	100
Subject objectives	"Software Engineering" course is aimed at explaining issues related to software development in industrial environment: complex systems designed for real customer/user, associated with particular business goals and expected level of quality, developed by large teams of software professionals.						

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_W42] Knows and understands, to an advanced extent, architecture, design principles and methods of hardware and software support for local and distributed information systems, including computing systems, databases, computer networks and information applications, as well as the principles of human cooperation with computers and computer-aided teamwork	The student understands the issues of IT systems design (on both architectural and module level), including software reuse and user interface design.	[SW1] Assessment of factual knowledge
	[K6_U03] can design, according to required specifications, and make a simple device, facility, system or carry out a process, specific to the field of study, using suitable methods, techniques, tools and materials, following engineering standards and norms, applying technologies specific to the field of study and experience gained in the professional engineering environment	The student develops analytical and design models of IT system using CASE (Computer Aided Software Engineering) software supporting tools.	[SU1] Assessment of task fulfilment
	[K6_W06] Knows and understands the basic processes occurring in the life cycle of devices, facilities and systems specific to a given field of study.	The student understands the importance of engineering practices and organisation of teamwork in software projects. Enumerates and describes key areas of software development process. Explains the selection of practices for the specific software project.	[SW1] Assessment of factual knowledge
	[K6_U43] can analyse data and formulate, apply and assess appropriate formal models and algorithms for solving problems in the field of information systems and applications	The student uses Unified Modeling Language to prepare the models of IT systems.	[SU1] Assessment of task fulfilment
	[K6_U09] can carry out a critical analysis of the functioning of existing technical solutions and assess these solutions, as well as apply experience related to the maintenance of technical systems, devices and facilities typical for the field of studies, gained in the professional engineering environment	The student develops "Vision of IT system" document that includes a critical analysis of the present state of the customer organization as well as basic requirements and restrictions of the planned IT system.	[SU1] Assessment of task fulfilment
Subject contents	<ol style="list-style-type: none"> <li>1. Introduction</li> <li>2. Scope and subject of software engineering. Essential motivations and concepts</li> <li>3. Areas of software engineering - an overview</li> <li>4. Planning and defining scope of software project. SSM approach and Rich Picture</li> <li>5. Risk and social responsibility related to IT systems</li> <li>6. Requirements engineering basics</li> <li>7. Conceptual modelling</li> <li>8. Use cases</li> <li>9. Object-oriented analysis using UML</li> <li>10. Modelling of logical system structure: class diagrams</li> <li>11. Modelling of system structure: other structural diagrams</li> <li>12. Modelling system dynamics: sequence and communication diagrams</li> <li>13. Modelling system dynamics: representing object's state</li> <li>14. System design: high-level design</li> <li>15. System design: class design (low level)</li> <li>16. Software reuse, design patterns</li> <li>17. User interface design: motivations, terms, techniques</li> <li>18. Software testing: terms, place in software development process</li> <li>19. Software testing: techniques (black/white box), levels of testing, managing tests</li> <li>20. Software deployment and maintenance</li> <li>21. Configuration management and software evolution</li> <li>22. Classical (waterfall) software lifecycle model</li> <li>23. Non-classical software lifecycles and development processes</li> <li>24. Adjusting development process to particular software project context</li> <li>25. Software development methodologies (plan-driven and agile)</li> </ol>		
Prerequisites and co-requisites	Presence during laboratory courses is mandatory. Delivery of all laboratory exercises and positive verification by tutor is required to pass the lab. Delays in delivering exercises affects the assessments. Only students who pass the lab are entitled to write the exam.		

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Written exam	50.0%	50.0%
	Lab (assignments & tests)	50.0%	50.0%
Recommended reading	Basic literature	1. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 2. Pressman R., Software Engineering: a Practitioner's Approach, 7th edition, McGraw-Hill, 2009 3. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010 4. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, 2005 5. Fowler M., UML distilled, 3rd edition, Addison-Wesley, 2003	
	Supplementary literature	No requirements	
	eResources addresses		
Example issues/ example questions/ tasks being completed			
Work placement	Not applicable		